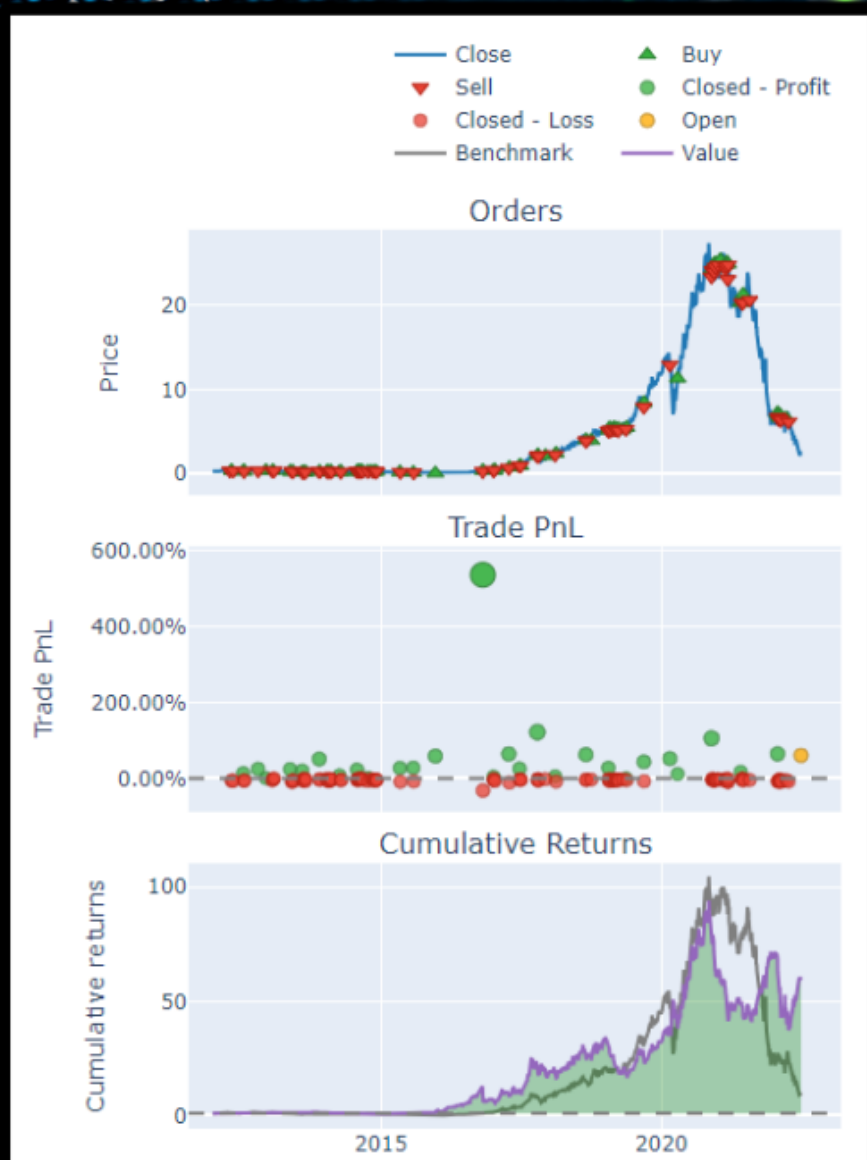


# VECTOR BT

Um guia prático para análise técnica



# INTRODUÇÃO



## Um guia prático para análise técnica

**Nos dias de hoje, em que os mercados financeiros são alimentados por dados, informações precisas e estratégias fundamentadas são vitais para o sucesso. A análise técnica, que se baseia no estudo de padrões de preços e indicadores derivados de dados de mercado, tornou-se uma ferramenta inestimável para os investidores e traders que buscam tomar decisões informadas. E agora, você tem a oportunidade de dominar essa habilidade por meio da linguagem de programação Python e da poderosa biblioteca VectorBT.**

# INTRODUÇÃO



**Este e-book foi cuidadosamente elaborado para os alunos da nossa escola de programação em Python, focando no emocionante cruzamento entre análise técnica e programação. Em particular, exploraremos detalhadamente os principais indicadores de análise técnica disponíveis na biblioteca VectorBT. Se você ainda não está familiarizado com a VectorBT, não se preocupe – durante este guia, forneceremos uma introdução abrangente a essa biblioteca e, em seguida, mergulharemos fundo nos indicadores técnicos essenciais que ela oferece.**



# VECTOR BT



## O que é?

**A biblioteca VectorBT é uma ferramenta poderosa e versátil projetada para capacitar os entusiastas do mercado financeiro e programadores de Python com as ferramentas necessárias para análise técnica avançada. Criada para simplificar e otimizar a análise de dados financeiros, a VectorBT oferece uma ampla gama de recursos que permitem aos usuários explorar e implementar estratégias de negociação de maneira eficaz.**

**Ela permite o cálculo e visualização de diversos indicadores, facilitando a interpretação de padrões, tendências e pontos de reversão.**



# BACKTESTING



## O que é?

O backtesting é um processo no qual uma estratégia de negociação é testada retrospectivamente usando dados históricos para avaliar como ela teria se saído no passado. É uma técnica essencial para avaliar a viabilidade e eficácia de estratégias de negociação antes de aplicá-las ao mercado em tempo real.

O backtesting envolve a execução da estratégia em dados históricos para simular negociações, calcular resultados financeiros e avaliar métricas de desempenho, permitindo que os traders identifiquem pontos fortes e fracos da estratégia e ajustem-na conforme necessário antes de implementá-la de fato.

# COMEÇANDO



## Instalando e importando

**Independente de qual plataforma de programação estiver utilizando, o procedimento de instalação da biblioteca é o abaixo:**



*#Instalação*

```
!pip install vectorbt
```

*#Importando a biblioteca*

```
import vectorbt as vbt
```

# DADOS



## Importando dados de cotações

**Para importar dados para seu backtesting a VectorBT conta com uma integração com a biblioteca YFinance, portanto a sintaxe é bem parecida:**

```
#Importando dados do Ibov
ibov = vbt.YFData.download('^BVSP', period='max',
interval='1d').get(['Open', 'High', 'Low', 'Close'])

#Importando dados de PETR4

petr4 = vbt.YFData.download('PETR4.SA', period='max',
interval='1d').get()
```



# INDICADORES



## Indicadores de análise técnica

**A VectorBT conta com diversos indicadores de análise técnica oriundos das bibliotecas pandas-ta e ta. Vamos criar alguns indicadores populares na análise técnica como objetos da Vector BT**

# INDICADORES



## Médias móveis

A média móvel é um indicador fundamental da análise técnica que suaviza os dados de preço ao longo de um determinado período, revelando a tendência subjacente de um ativo financeiro. Ela é calculada somando os preços de fechamento de um ativo durante um intervalo de tempo específico e dividindo esse total pelo número de períodos. Isso resulta em uma linha que acompanha as flutuações dos preços, tornando as tendências mais claras e ajudando a identificar reversões ou movimentos significativos.



```
#Média móvel de 20 períodos do preço de fechamento de PETR4
```

```
MM_20p = vbt.MA.run(petr4.Close, 20).ma
```

```
#Trazendo o indicador para dentro do dataframe de OHLC
```

```
petr4['MMA20'] = MM_20p
```

# INDICADORES



## Relative Index Strength

O Índice de Força Relativa (RSI) é um indicador entre 0 e 100 que mede a rapidez das mudanças nos preços de um ativo. Ele ajuda a identificar momentos de sobrecompra (RSI acima de 70) e sobrevenda (RSI abaixo de 30), indicando possíveis reversões de tendência. O RSI oferece insights sobre a força das tendências e auxilia na tomada de decisões de negociação.



```
#RSI do preço de fechamento de PETR4 janela de 14 períodos  
ifr = vbt.RSI.run(petr4.Close, window=14).rsi  
  
#Trazendo o indicador para dentro do dataframe de OHLC  
petr4['RSI'] = ifr
```



# INDICADORES



## Bandas de Bollinger

Um indicador que consiste em uma linha central (média móvel) cercada por duas faixas, representando desvios padrão dos preços. Essas bandas se expandem ou contraem com a volatilidade do mercado. Quando os preços se aproximam das bordas superiores ou inferiores das bandas, pode indicar condições de sobrecompra ou sobrevenda, sugerindo possíveis reversões de tendência.



```
#Bandas de Bollinger com janela de 16 períodos e 2 desvios  
bbol_ativo = ta.volatility.BollingerBands(dados_ohlc.Close,  
window=16, window_dev=2)
```

```
#Trazendo o indicador para dentro do dataframe de OHLC
```

```
dados_ohlc['BBol_sup'] = bbol_ativo.bollinger_hband()  
dados_ohlc['BBol_inf'] = bbol_ativo.bollinger_lband()  
dados_ohlc['BBol_media'] = bbol_ativo.bollinger_mavg()
```

# SINAIS



## Sinais de entrada e saída

Na Vector BT para orientar as entradas e saídas da estratégia é necessário criar os sinais dentro da biblioteca



```
#Capturando os dados
vale = vbt.YFData.download('VALE3.SA', start='2012-01-01',
end='2022-06-30').get('Close')

#Calcular a média móvel
MMA50p = vbt.MA.run(vale, 50, short_name='MMA').ma

#Definir regras de entradas e saídas
entradas = vale.vbt.crossed_above(MMA50p)
saidas = vale.vbt.crossed_below(MMA50p)
```

Nesse exemplo estamos criando uma estratégia com uma média móvel onde teremos entrada assim que o preço de fechamento cruzar acima da MM50 períodos e uma saída com ele cruzar abaixo da MM50.

# BACKTESTING



## Fazendo o backtesting

Para construir o backtesting dentro da Vector BT é bem simples:



```
#Construindo Backtest com fees de 0.005% por operação operando  
#comprado e vendido (direction)
```

```
mm50_vale = vbt.Portfolio.from_signals(vale,  
                                       entradas,  
                                       saidas,  
                                       fees=0.005,  
                                       size=np.inf,  
                                       direction='both',  
                                       freq='D')
```

O primeiro parâmetro será seu dataframe e dados open high low e close, os sinais (entrada e saída), taxas se aplicável (fees) e direction pode ser longonly (apenas compra), shortonly (apenas venda), both (os dois).



# RESULTADOS



## Analizando resultados

A Vector BT conta com alguns métodos para analisar o resultado do seu backtesting, o principal é o `.stats` e o `.plot` para avaliar as entradas e gains



```
#Métricas
```

```
mm50_vale.stats()
```

```
#Gráfico de performance
```

```
mm50_vale.plot(width=500, height=700).show()
```

# RESULTADOS



**.stats**

**Esse método traz diversas métricas estatísticas e financeiras sobre o desempenho da sua estratégia poupando bastante tempo montando as análises**

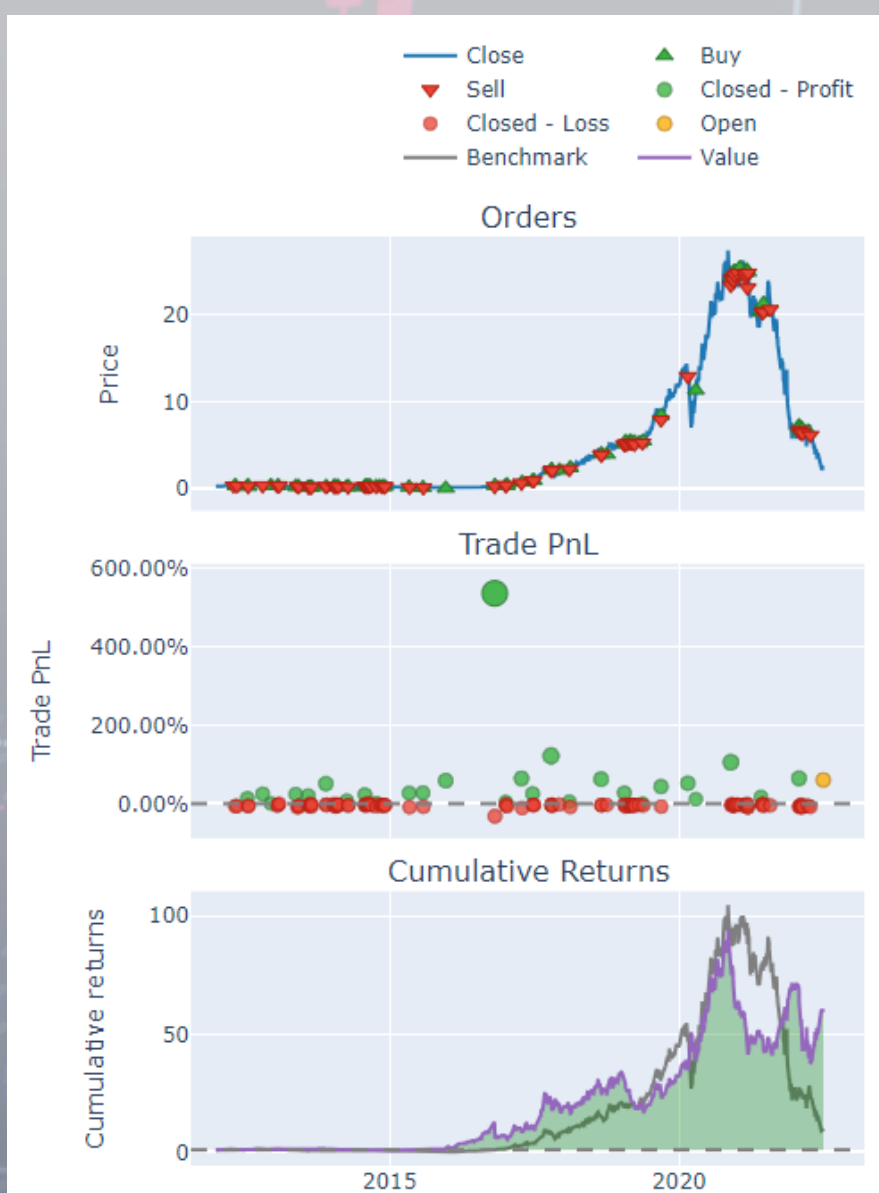
Start	2012-01-02 00:00:00+00:00
End	2022-06-29 00:00:00+00:00
Period	2602 days 00:00:00
Start Value	100.0
End Value	6024.77846
Total Return [%]	5924.77846
Benchmark Return [%]	820.357535
Max Gross Exposure [%]	100.0
Total Fees Paid	2726.227397
Max Drawdown [%]	61.86107
Max Drawdown Duration	523 days 00:00:00
Total Trades	139
Total Closed Trades	138
Total Open Trades	1
Open Trade PnL	2256.600635
Win Rate [%]	19.565217
Best Trade [%]	535.091943
Worst Trade [%]	-32.075414
Avg Winning Trade [%]	52.536041
Avg Losing Trade [%]	-4.520962
Avg Winning Trade Duration	72 days 09:46:40
Avg Losing Trade Duration	4 days 15:34:03.243243243
Profit Factor	1.343699
Expectancy	26.580999
Sharpe Ratio	1.238231
Calmar Ratio	1.25602
Omega Ratio	1.213756
Sortino Ratio	1.881464

# RESULTADOS



**.plot**

**Gráficos de desempenho da estratégia, profit and loss e retornos acumulados da estratégia**





# CONCLUSÃO



**Chegamos ao fim!**

**À medida que concluimos esta jornada pelo e-book de Análise Técnica com VectorBT, você agora possui uma sólida compreensão de como a linguagem de programação Python e a poderosa biblioteca VectorBT podem ser aliadas na sua busca por dominar a análise técnica no mercado financeiro. Você aprendeu a instalar a biblioteca, uma etapa crucial para explorar suas funcionalidades ricas, e adquiriu habilidades práticas na construção de estratégias de negociação usando cruzamento de médias móveis. estratégia**

# CONCLUSÃO



**Chegamos ao fim!**

**Além disso, você mergulhou na interpretação de resultados, descobrindo como analisar os efeitos de suas estratégias e ajustá-las conforme necessário. Este e-book serviu como seu guia nessa exploração do casamento entre programação e análise técnica, capacitando-o a tomar decisões informadas e desenvolver abordagens sólidas no mercado financeiro em constante evolução. Parabéns por seu compromisso com o aprendizado e sua jornada em direção a uma compreensão mais profunda dos mercados e suas oportunidades. Continue a aplicar esse conhecimento e aprimorar suas habilidades, lembrando sempre que a análise técnica é uma ferramenta poderosa, mas a adaptação constante e a busca por insights são essenciais para prosperar no emocionante mundo das finanças.**